بسمه تعالی

تمرین سری پنجم درس یادگیری عمیق

پوریا محمدی نسب

۴۰۰۷۲۲۱۳۸

فهرست

سوال ۱.

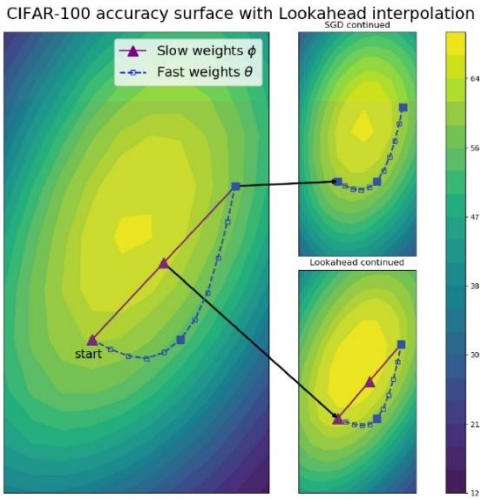
مقاله زیر را با دقت مطالعه بفرمایید و ایده کلی آن برای بهینه سازی را همراه با مزایای آن توضیح دهید.

Zhang, Michael & Lucas, James & Hinton, Geoffrey & Ba, Jimmy *"Lookahead Optimizer: k steps forward, 1 step back"* NeurIPS Proceedings, (2019).

به طور کلی در سال های اخیر بهبود الگوریتم بهینه ساز SGD از دو جهت مورد توجه قرار گرفته است. دسته ی اول مقالاتی هستند که سعی دارند تا نرخ آموزشی(learning rate) الگوریتم را پویا (adaptive) کنند. دسته دوم الگوریتم هایی هستند سعی بر تسریع عمل بهینه سازی دارند. اما الگوریتم معرفی شده در مقاله مورد نظر سوال در روشی است که برای بهبود عمل بهینه سازی از روش های دو دسته ذکر شده استفاده نمیکند.

در این الگوریتم ۲ مجموعه وزن را در نظر میگیریم که شامل مجموعه وزن های کند (Φ) و مجموعه وزن های سریع(θ) هستند. که هر k تا آپدیت باعث هماهنگ سازی مجموعه وزن ها میشود. مجموعه وزن های سریع با اعمال k بار یک الگوریتم بهینه سازی استاندارد بر روی یک Batch از دیتاست آپدیت میشوند. پس از این k بار نوبت به آپدیت وزن های کند میرسد که با یک رابطه ی خطی (θ - Φ) افزایش میابند. نرخ یادگیری وزن های کند را با α نمایش میدهیم. پس از اپدیت کردن وزن های کند، وزن های سریع به مقدار وزن های کندی که جدیدا بدست آمده تغییر میکنند.

شکل زیر، روند مصور الگوریتم و شبه کد الگوریتم را نمایش میدهد که در بالا توضیح داده شد.



Algorithm 1 Lookahead Optimizer:

Require: Initial parameters ϕ_0, objective function L
Require: Synchronization period k, slow weights step size α, optimizer A
for $t = 1, 2, \ldots$ **do**
  Synchronize parameters $\theta_{t,0} \leftarrow \phi_{t-1}$
  for $i = 1, 2, \ldots, k$ **do**
    sample minibatch of data $d \sim \mathcal{D}$
    $\theta_{t,i} \leftarrow \theta_{t,i-1} + A(L, \theta_{t,i-1}, d)$
  end for
  Perform outer update $\phi_t \leftarrow \phi_{t-1} + \alpha(\theta_{t,k} - \phi_{t-1})$
end for
return parameters ϕ

در ادامه به بررسی نتایج این روش میپردازیم. در ابتدا عملکرد این بهینه ساز بر روی دیتاست های معروف تصویری CIFAR 10/100 میپردازیم. ۴ بهینه سازی شامل SGD، POLYAK، ADAM و LookAhead بر روی این دو دیتاست استفاده شدند و نتایج دقت و واریانس آنها در جدول زیر آمده است. نکته مهم این است که بهینه ساز LA علاوه بر داشتن دقت قابل قبول، از پایداری بالاتری (واریانس کمتری) برخوردار است.

OPTIMIZER	CIFAR-10	CIFAR-100
SGD	$95.23 \pm .19$	$78.24 \pm .18$
POLYAK	$95.26 \pm .04$	$77.99 \pm .42$
ADAM	$94.84 \pm .16$	$76.88 \pm .39$
LOOKAHEAD	$95.27 \pm .06$	$78.34 \pm .05$

دیتاست بعدی imagenet است که یکی از پیچیده ترین دیتاست ها برای هر مدل یادگیری است. عملکرد الگوریتم های LA و SGD را تحت شرایط مختلف از جمله تعداد epoch متفاوت و اندازه گیری دقت با معیار های مختلف در جدول زیر مشاهده میکنید. معیارهای متفاوت اندازه گیری دقت در جدول زیر Top-1 accuracy و Top-5 accuracy هستند که به ترتیب وقتی پیشبینی مدل درست در نظر گرفته میشود اگر امتیاز شی مورد نظر بهترین با جزء ۵ تا از بهترین امتیازات باشد.

Optimizer	LA	SGD
Epoch 50 - Top 1	75.13	74.43
Epoch 50 - Top 5	92.22	92.15
Epoch 60 - Top 1	75.49	75.15
Epoch 60 - Top 5	92.53	92.56

سوال ۲.

۲- الف)

با توجه به صفحه ۴۵۵ - ۴۵۹ کتاب، روش نیوتون را توضیح دهید، و بیان کنید مزیت اینکه این روش از ۲ ترم اول سری تیلور استفاده می کند در چیست؟

روش نیوتون یکی از روش های وفقی (Adaptive) برای بهینه سازی است. اساس کار این روش این است که از در بسط تیلور ار مرتبه دوم نیز استفاده کنیم. پس رابطه آن به صورت زیر است:

$$f(x+\varepsilon) = f(x) + \varepsilon^T \nabla^2 f(x) + \frac{1}{2}\,\varepsilon^T \nabla^2 f(x)\varepsilon + O\left(||O^3||\right)$$

که در واقع ∇^2 یک ماتریس d×d است که به ماتریس Hessian معروف است و با H_f نمایش میدهیم. البته باید توجه کرد که محاسبه ی ماتریس Hessian برای d های کوچک مناسب است اما برای ابعاد بالا هزینه بسیار بالای محاسباتی و حافظه ای دارد. در این روش برای یافتن مقدار بهینه تابع باید مشتق آن نسبت به ε مشتق میگیریم و برابر صفر قرار میدهیم.

$$\nabla f(x) + H\varepsilon = 0 \;\rightarrow\; \varepsilon = \; -H^{-1}\nabla f(x)$$

نکته ای که از رابطه بالا متوجه میشویم این است که معکوس ماتریس Hessian به جای نرخ یادگیری در فرآیند یادگیری استفاده میشود. مزیت این که از ۲ ترم اول سری تیلور استفاده میکنیم این است که تقریب مرتبه دوم ۲ دارای یک بهینه سراسری و قابل محاسبه است.

۲- ب)

توضیح دهید در چه شرایطی روش نیوتون با مشکل مواجه میشود و را ه حلهای آن را بیان کنید.

مشکلات:

۱) در قسمت الف سوال نیز توضیح دادم که مقدار d از حدی بیشتر شود از لحاظ محاسباتی و حافظه مشکلات جدی داریم.

۲) اگر تابعی که قصد بهینه سازی آن را داریم convex نباشد در انتهای الگوریتم که مقدار بدست آمده را به مقدار متناظر Hessian تقسیم میکنیم با این کار جایی که مشتق دوم منفی است ما به سمت افزایش f میرویم و این باعث میشود الگوریتم کاملا اشتباه کار کنه.

راه حل:

۱) یک روش ثابت کردن مقادیر Hessian با مقادیر مطلق آن است.

۲) یک روش دیگر کاهش دادن مقدار نرخ آموزش است.

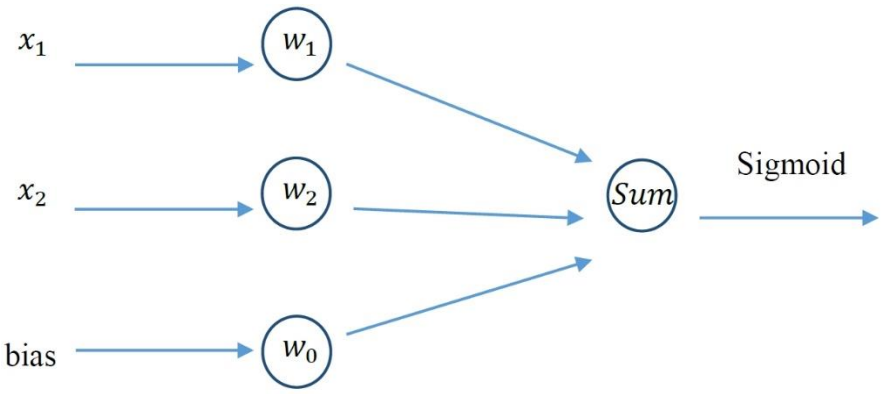
سوال ۳.

به نظر شما، در روش SGD با batch_size = 1 وقتی از $\{(x1,y1),...,(xn,yn)\}$ با جایگزینی و بدون جایگزینی نمونه بردار ی شود، چه تفاوتی در عملکرد بهینه ساز ایجاد میشود ؟

روش SGD برای انتخاب sample از مجوعه آموزشی از روش uniform sampling استفاده میکند به این معنا که همه ی batch ها شانس برابری برای انتخاب شدن دارند. در سوال ما، bath-size=1 است پس هر instance به تنهایی یک batch حساب میشود. در این حال میتوان از انتخاب با جایگزینی با بدون جایگزینی استفاده کرد. در حالتی که از جایگزینی استفاده کنیم اجازه میدهیم در یک mini-batch ، یک سمپل بیشتر از یک با هم وجود داشته باشد اما در حالت بدون جایگزینی این اجازه را به sample ها نمیدهیم و هر sample حداکثر یک بار میتواند در mini-batch وجود داشته باشد. با توجه به صورت سوال که اندازه هر batch یک است تفاوتی ندارد که از حالت با جایگزینی یا بدون جایگزینی استفاده کنیم.

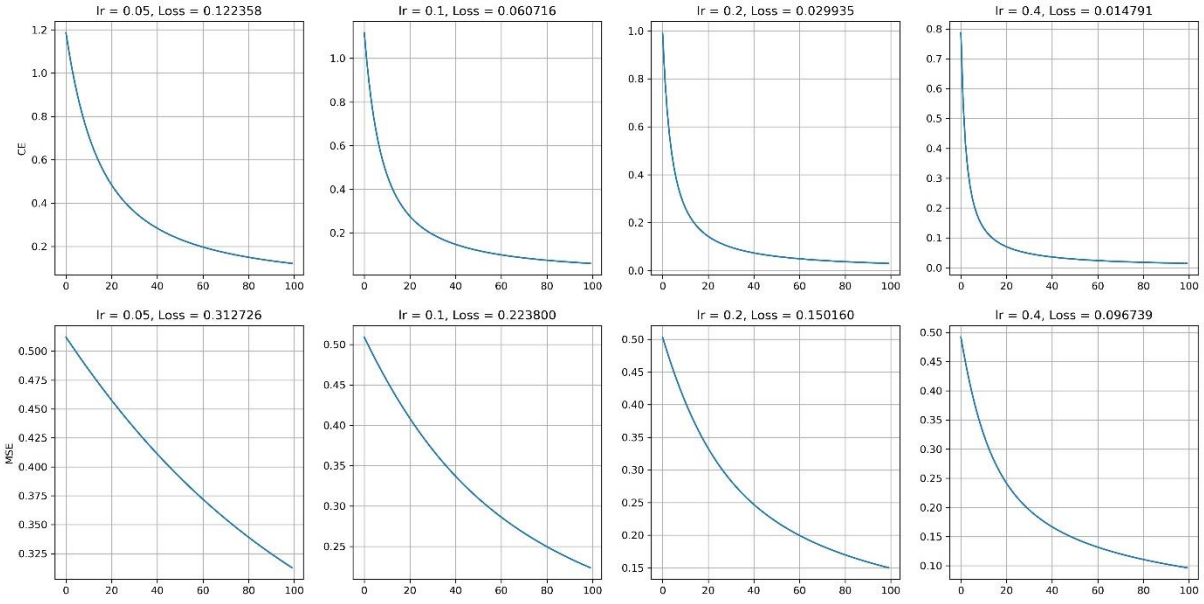
سوال ۴.

شبکه ساده زیر را در نظر بگیرید. م یخواهیم با استفاده از شبکه زیر یک مسئله دو کلاسه را حل کنیم .



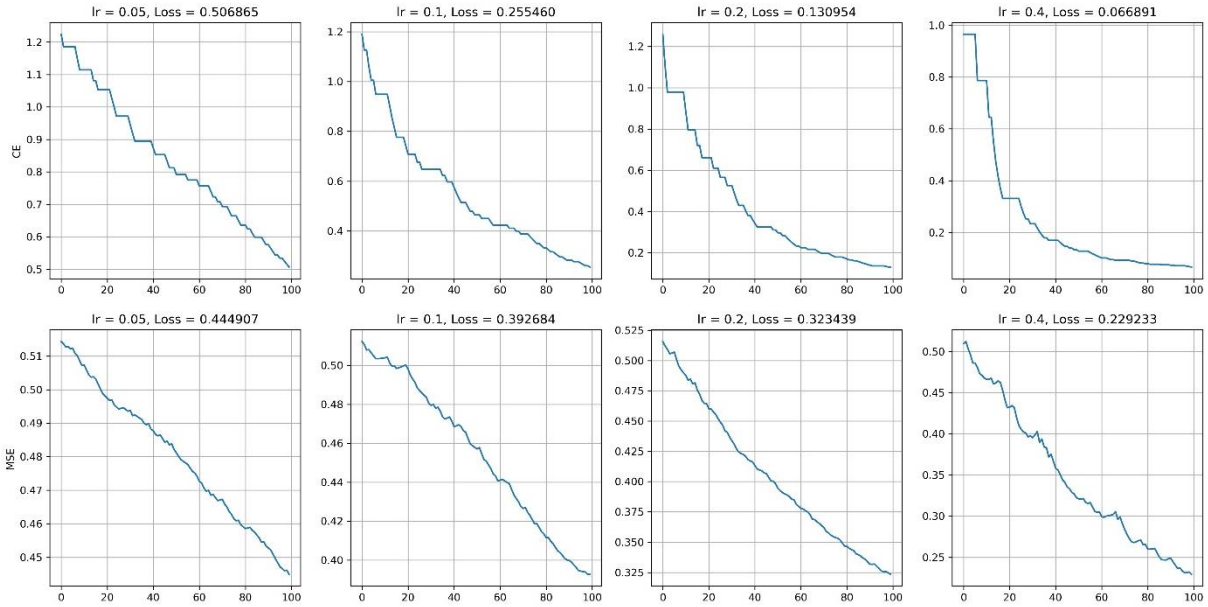
۴- الف) GD

شکل زیر مقایسه ای جامع از پیاده سازی الگوریتم GD را نشان میدهد که با ۴ نرخ آموزش مختلف و دو نوع خطای MSE و CE محاسبه شده است. نتایج نشان میدهد که خطای نوع CE مناسب این سوال کلاسبندی است و MSE نتایج خوبی ندارد و همینطور هر چه نرخ آموزش در این مثال خاص بیشتر شود نتایج بهتر میشود البته این مقدار افزایش حد آستانه نیز دارد.



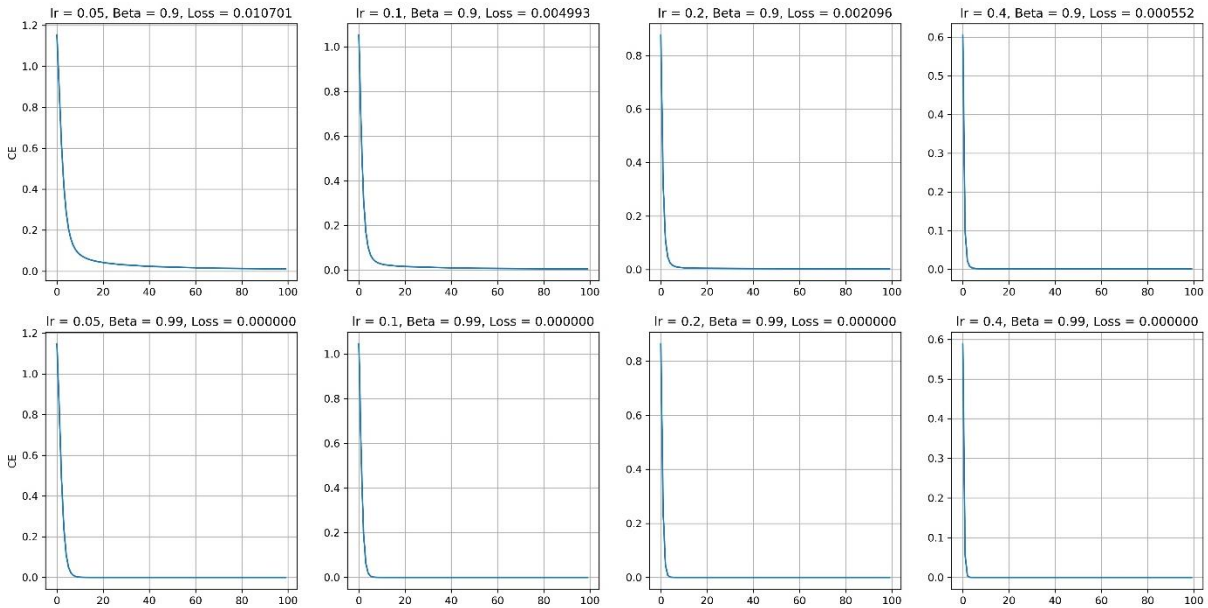
۲ – ب) SGD با batch_size = 1

همانند GD در SGD نیز خطای CE مناسب ترین نوع loss است.

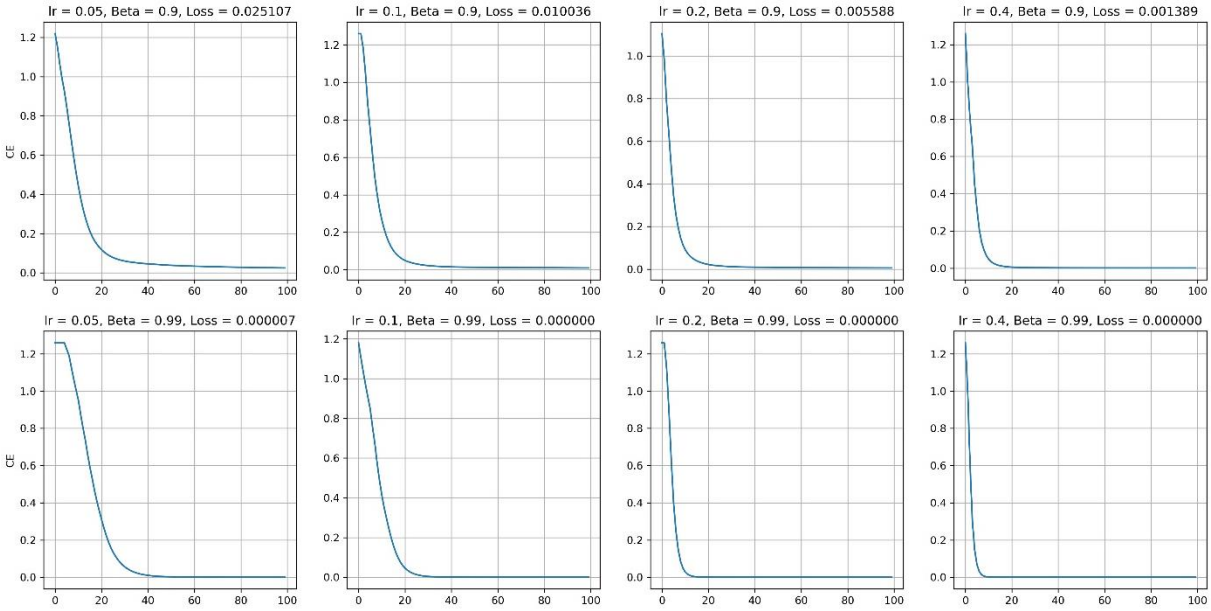


۴- پ) GD + Momentum

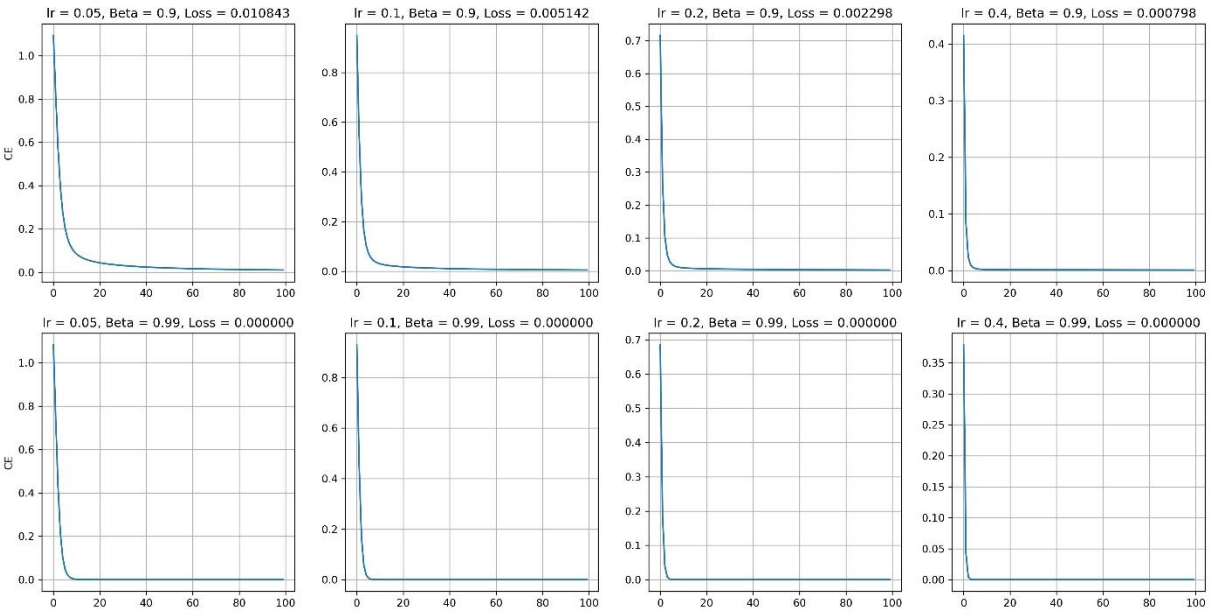
برای مقایسه β های مختلف (0.9 و 0.99) برای اندازه گیری loss فقط از متد Cross entropy استفاده کردیم. ردیف بالای نمودارهای زیر با β = 0.9 و ردیف دوم با β = 0.99 و نرخ های آموزشی مختلف هستند.



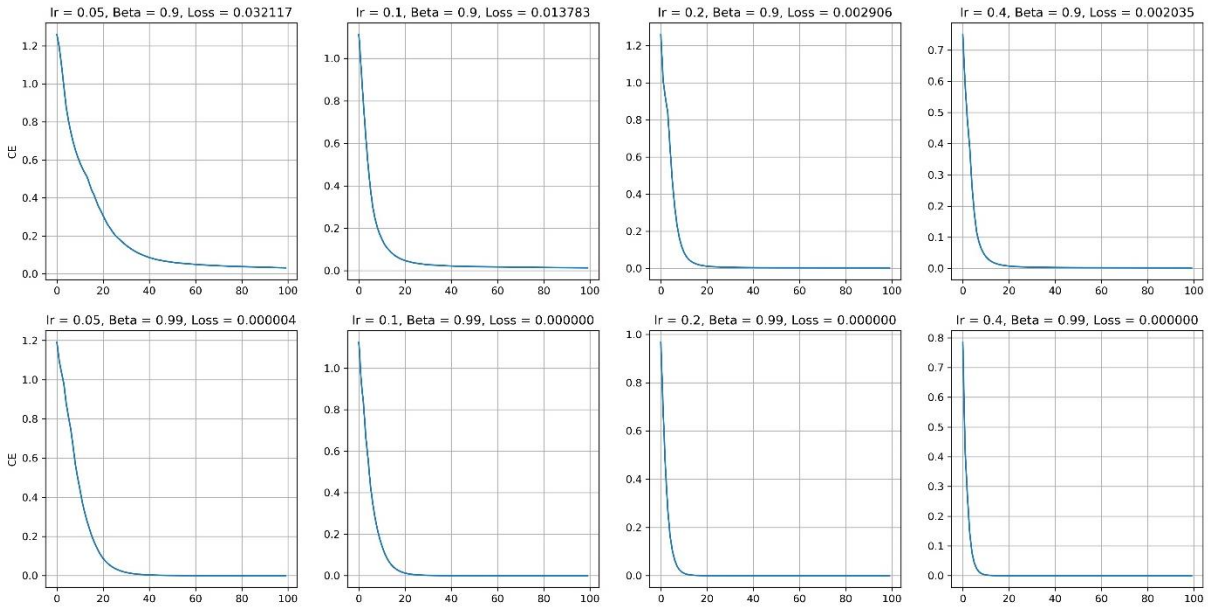
۴ – ت) SGD + Momentum



۴ – پ) GD + Nesterov Momentum



۴ – ت) SGD + Nesterov Momentum



مراجع

1) https://classic.d2l.ai/chapter_optimization/minibatch-sgd.html

2) https://proceedings.neurips.cc/paper/2019/file/90fd4f88f588ae64038134f1eeaa023f-Paper.pdf